I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail in an envelope addressed to:

ASSISTANT COMMISSIONER OF PATENTS
WASHINGTON, DC 20231

bearing Label Number EL 057 650 493 US and mailed July 31, 2001.

Ira Richardson
Print Name

Ira Richardson
Signature

Patent

Inventors: Richard A. Dayan
Joseph W. Freeman
William F. Keown
Randall S. Springfield

RECOVERY OF A BIOS IMAGE

TECHNICAL FIELD

The present invention relates in general to recovering the Basic Input Output System (BIOS) image for a computer system upon a remote or local request or in the case of a boot-up failure.

BACKGROUND INFORMATION

The BIOS image, which includes both programs and data, is an essential set of routines in a personal computer (PC) or other computer system which is stored within the computer and provides an interface between the operating

system and the hardware. The read-only locations in the BIOS image contain boot block code that is executed first after any power-up or system reset of the computer system. The BIOS image supports all peripheral technologies and internal services such as the real-time clock (time and date). On startup, code in the BIOS image (may be referred to as simply BIOS) tests the system and prepares the computer for operation by querying its own small memory bank for peripheral drive and other configuration settings. It searches for other BIOS images on the plug-in boards and sets up pointers (interrupt vectors) in memory to access those routines. It then loads the operating system and passes control to it. The BIOS accepts requests from the peripheral drivers as well as the application programs. The BIOS image must periodically be updated to keep pace with new peripheral technologies. If the BIOS image is stored in a read-only memory (ROM) chip (ROM BIOS), then to update the BIOS image the ROM chip must be replaced. In newer systems, the BIOS image is stored on an electronically erasable programmable read only memory (EEPROM) that may be upgraded via software. One problem with this method is that the BIOS in the EEPROM may be corrupted during the upgrade process. For example, corruption may occur if power is lost while updating the system BIOS image. To recover from the power loss, the covers of the system are opened and a jumper directs the receipt of a new BIOS image from an external storage device (e.g., a diskette). The diskette data is recorded into the EEPROM and then upon system re-boot the new BIOS image is used for setting up the system. The above method requires the system to be equipped with a diskette or other external readable storage media (e.g., CD-ROM) and it requires that the system covers be opened to manually direct (e.g., with a jumper) writing of the EEPROM with new BIOS image. Many new systems, while containing a hard drive, come without a diskette, CD-ROM, or other removable media drive. These systems which contain only a hard drive still require some method to recover a lost or corrupted BIOS image. It would be convenient to not require the opening of the

covers of a system to update the stored BIOS image in an EEPROM, and it would be cost effective to eliminate nonessential storage devices incorporated in systems primarily for BIOS image recovery.

Therefore, there is a need for a method and system to allow the BIOS image to be recovered in a system with only a hard drive without requiring the covers of the system to be opened or requiring that the system be equipped with a diskette drive or a CD-ROM drive.

SUMMARY OF THE INVENTION

A computer system has an Integrated Drive Electronics (IDE) compatible hard drive storage device and a method for communicating with the system via either a wide area network (WAN) connection or a local area network (LAN). The IDE or equivalent hard drive is equipped with a feature that supports designations of partitions of the hard drive to be hidden and protected during the period when the operating system (OS) is in control and executing. The boot block code in the BIOS image stored in the EEPROM includes recovery code that searches the hidden partition of the hard drive for a BIOS image. The system manufacturer may place a recovery image in the hidden partition which then is used to recover the system in case the original BIOS image in the EEPROM is corrupted. If the active BIOS image in the EEPROM is corrupted, then on a recovery event, the boot block recovery code rewrites the BIOS image in the EEPROM with the BIOS image stored in the hidden portion of the hard drive. The BIOS image in the hidden partition of the hard drive may be updated via the external communication connection or in some instances from a diskette drive or a compact disk read-only memory (CD-ROM) drive.

The foregoing has outlined rather broadly the features and technical advantages of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter which form the subject of the claims of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram of a computer with an EEPROM for storing a BIOS image and a CD-ROM or diskette drive for updating the BIOS image;

FIG. 2 is a block diagram of a computer system according to embodiments of the present invention;

FIG. 3 is a flow diagram of steps in a method according to embodiments of the present invention; and

FIG. 4 is a data processing system configured to use embodiments of the present invention for recovering or updating the BIOS image.

DETAILED DESCRIPTION

In the following description, numerous specific details are set forth to provide a thorough understanding of the present invention. However, it will be obvious to those skilled in the art that the present invention may be practiced without such specific details. In other instances, well-known circuits have been shown in block diagram form in order not to obscure the present invention in unnecessary detail. For the most part, details concerning timing considerations and the like have been omitted in as much as such details are not necessary to obtain a complete understanding of the present invention and are within the skills of persons of ordinary skill in the relevant art.

Refer now to the drawings wherein depicted elements are not necessarily shown to scale and wherein like or similar elements are designated by the same reference numeral through the several views.

Integrated Drive Electronics (IDE) is a type of hardware interface widely used to connect hard disks, CD-ROMs and tape drives to a personal computer (PC). IDE, as an interface, is very popular because it is an economical way to connect peripherals. Starting out with 10MB capacities years ago, 6GB and 10GB IDE hard disks have become entry level, costing less than a penny per megabyte. With IDE, the controller electronics is built into the drive itself, requiring a simpler circuit in the PC for connection. IDE drives were attached to earlier PCs using an IDE host adapter card. Today, two Enhanced IDE (EIDE) sockets are built onto the motherboard, and each socket connects to two devices via a 40-pin ribbon cable.

FIG. 1 is a block diagram of a computer system with an electronically erasable read-only memory (EEPROM) 102, a read/write magnetic disk storage (hard drive) 103, a diskette drive 104 for diskette media, and a compact disk read-only memory (CD-ROM) drive 105. The hard drive 103 and the diskette drive 104 may be used to store data and programs for the computer system.

CD-ROM 105 is typically used for loading programs or other data into computer system 101. EEPROM 102 may be used to store a BIOS image used by computer system 101 during boot-up to set the proper states of various I/O devices and hardware within computer 101. If the BIOS for computer system 101 gets corrupted, a diskette or CD-ROM may be used to update the BIOS image in EEPROM 102. Since the computer system 101 would typically read BIOS information from EEPROM 102 during a re-boot, a wire jumper is sometimes used to switch computer system 101 to store a new BIOS image from diskette drive 104 or CD-ROM drive 105 in EEPROM 102 for future re-boot operations. To change a wire jumper connection for this operation typically requires removing the covers of computer system 101.

FIG. 2 is a block diagram of computer system 201 without a CD-ROM drive or a diskette drive 104. System 201, in this embodiment of the present invention, uses an external connection 205 to allow a new BIOS image to be loaded from a hidden partition 204 on IDE hard drive 203. A normal re-boot or recovery of the BIOS image accesses EEPROM 202. However, if the BIOS image in the EEPROM 202 has been corrupted, then a normal re-boot would fail. A re-boot normally occurs as the result of a power-up instigated locally at the system or via a remote link. The system supports the "Wake on LAN" (WOL) protocol for remotely requesting a power-up of the system. When a power-up occurs, the boot block code accesses registers to determine the source of the power-up command. In this way, the boot block code may start a recovery process where the BIOS image in the EEPROM 202 is checked for validity. If the BIOS image in EEPROM 202 is corrupt, then the new BIOS image in the hidden partition 204 is checked for validity. If the new BIOS image is valid, it is written into EEPROM 202 in place of the existing BIOS image. If the boot block code determines that a WOL was used to issue a system power-up, then it instigates a BIOS recovery process if the WOL is properly authenticated. EEPROM 202 has recovery code that verifies the WOL packet against a known

signature. If the WOL is authenticated, then the present BIOS image EEPROM 202 may be rewritten with the new BIOS image from the hidden partition 204 on hard drive 203 and system may re-boot using the BIOS image from EEPROM 202.

FIG. 3 is a flow diagram of method steps in an embodiment of the present invention. In step 301, a power on request (POR) is received. In step 302, a register in system 201 is checked to determine the source of the POR. A test is done in step 303 to determine if the POR request is the result of a local POR in system 201. If the result of the test in step 303 is NO, then a test is done in step 304 to determine if the remote request via external communication link 205 request is valid. Communication over external communication link 205 may be secured by a variety of techniques which enable the boot block code to determine if the remote request is valid. If the result of the test in step 304 is NO, then an invalid request is Flagged and the request is ignored in step 305. If the result of the test in step 304 is YES, then the signature (e.g., a check sum) of the BIOS image is validated in step 307 to determine if the EEPROM 202 BIOS image has been corrupted. If the result of the test in step 307 is NO, then in step 313 a test is done to determine if the WOL request still requires that the BIOS in EEPROM 202 be updated or the jumper has been set to force an update in the case of a local POR. If the result of the test in step 313 is NO, then in step 312 a normal boot-up is executed using the present EEPROM BIOS 202 image. If the result of the test in step 313 is YES, then in step 308, the hidden partition 204 of IDE drive 203 is unlocked by the boot block code and the new BIOS image is loaded. If the result of the test in step 307 is YES, then the present EEPROM 202 BIOS image has been corrupted and a new BIOS image needs to be used for boot-up. In this case, step 308 is executed as before. In step 309, the signature of the new BIOS image in EEPROM 202 is checked to determine if the new BIOS image is valid. If the result of the test in step 309 is NO, then the boot-up is Halted as there is no valid BIOS image to

boot-up the system. If the result of the test in step 309 is YES, then in step 310 the new BIOS image is written into EEPROM 202. In step 312, the system is booted up normally using the new BIOS image written into EEPROM 202.

FIG. 4 is a high level functional block diagram of a representative data processing system 400 suitable for practicing the principles of the present invention. Data processing system 400, includes a central processing system (CPU) 410 operating in conjunction with a system bus 412. System bus 412 operates in accordance with a standard bus protocol, such that as the ISA protocol, compatible with CPU 410. CPU 410 operates in conjunction with electronically erasable programmable read-only memory (EEPROM) 416 and random access memory (RAM) 414. Among other things, EEPROM 416 supports storage for the Basic Input Output System (BIOS) data and recovery code. RAM 414 includes, DRAM (Dynamic Random Access Memory) system memory and SRAM (Static Random Access Memory) external cache. I/O Adapter 418 allows for an interconnection between the devices on system bus 412 and external peripherals, such as mass storage devices (e.g., an IDE hard drive, floppy drive or CD/ROM drive), or a printer 440. A peripheral device 420 is, for example, coupled to a peripheral control interface (PCI) bus, and I/O adapter 418 therefore may be a PCI bus bridge. User interface adapter 422 couples various user input devices, such as a keyboard 424, mouse 426, touch pad 432 or speaker 428 to the processing devices on bus 412. Display 439 which may be, for example, a cathode ray tube (CRT), liquid crystal display (LCD) or similar conventional display units. Display adapter 436 may include, among other things, a conventional display controller and frame buffer memory. Data processing system 400 may be selectively coupled to a computer or telecommunications network 441 through communications adapter 434. Communications adapter 434 may include, for example, a modem for connection to a telecom network and/or hardware and software for connecting to a computer network such as a local area network (LAN) or a wide area

network (WAN). CPU 410 may be a processor system employing the recovery of a BIOS image according to embodiments of the present invention. CPU 410 may also be operable to execute instructions implementing method steps according to embodiments of the present invention.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the spirit and scope of the invention as defined by the appended claims.